Name	Reg#No
Ahmad Ashayer	12028561
Mahmoud Arafat	12027669

Dos Project Part 1 Documentation

For this project we built it in express using nodejs, we have three containers which are:-Front, Catalog, and Order.

To run the project we used docker and with docker files such as the docker file itself and docker-compose.yml the project will run into Ubuntu operating system.

And by running the docker file it will create 3 containers and install all packages needed to run this project
For Run the project we first need to run two commands which are:-

docker-compose build

docker-compose up

The first one for building the packages after creating the images and the containers needed like express,sqlite3,Ubuntu,…and another needed things.

After that the second command is for running each container which hare three containers: **Front, Catalog, and Order.**

Each one runs on it’s port which are:- **3004,3003, and 3002** in order.

After running these two commands above, we’ll need to make the request call’s which are:- **info, search, and purchase.**

As this projects built as a micro service, the **Front** container has the possibility to router or sends all the request we need to do for the other containers like **Catalog** and **Order,** and we'll use port **3004** for the **Front**.

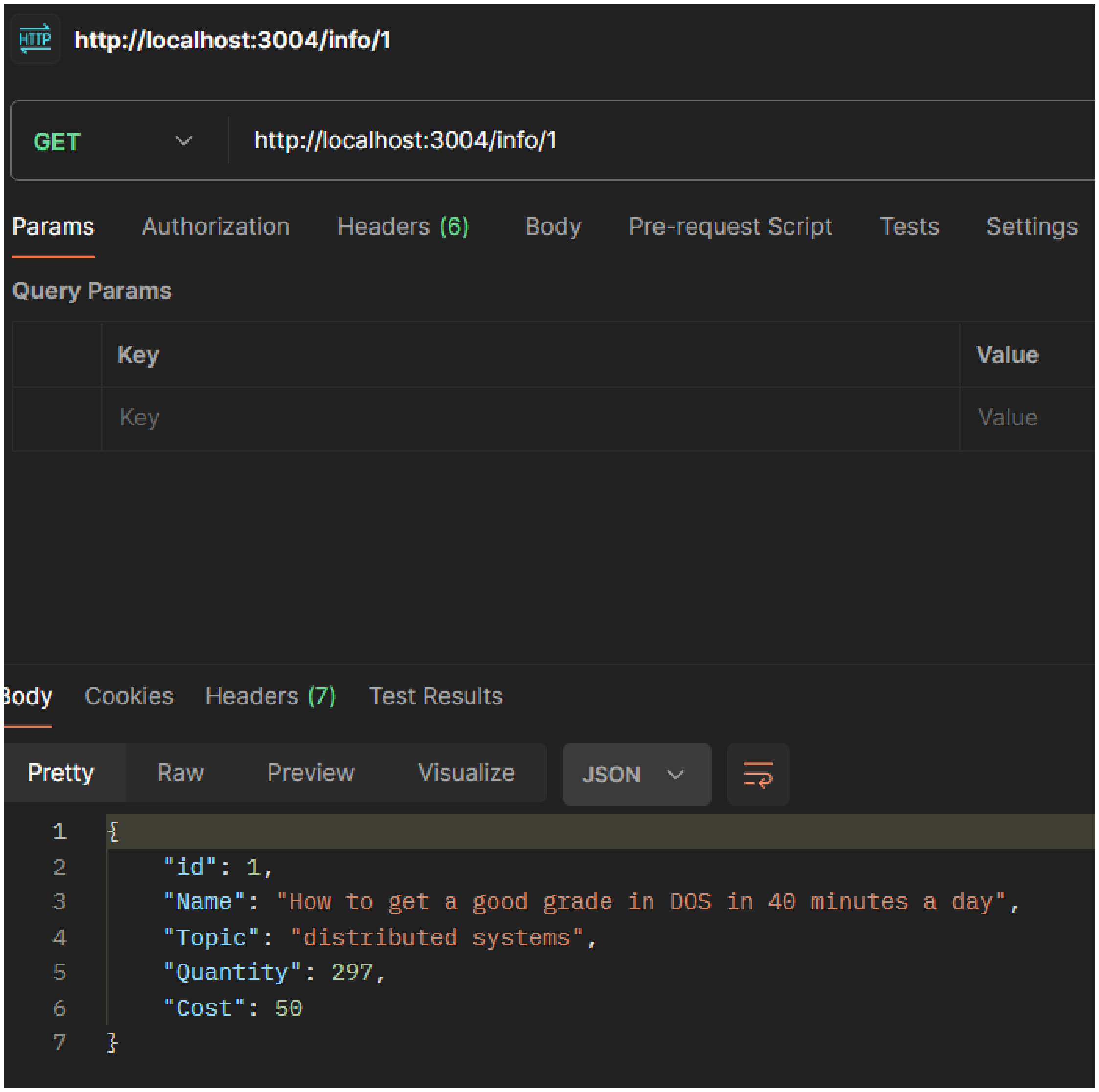
And also the order passes it's requests to Catalog, All of this done by using **Axios**.

We used postman for test each request with it's response.

1-To get information about any book, send

http://localhost:3004/info/{id}

this is an example with it's output:-



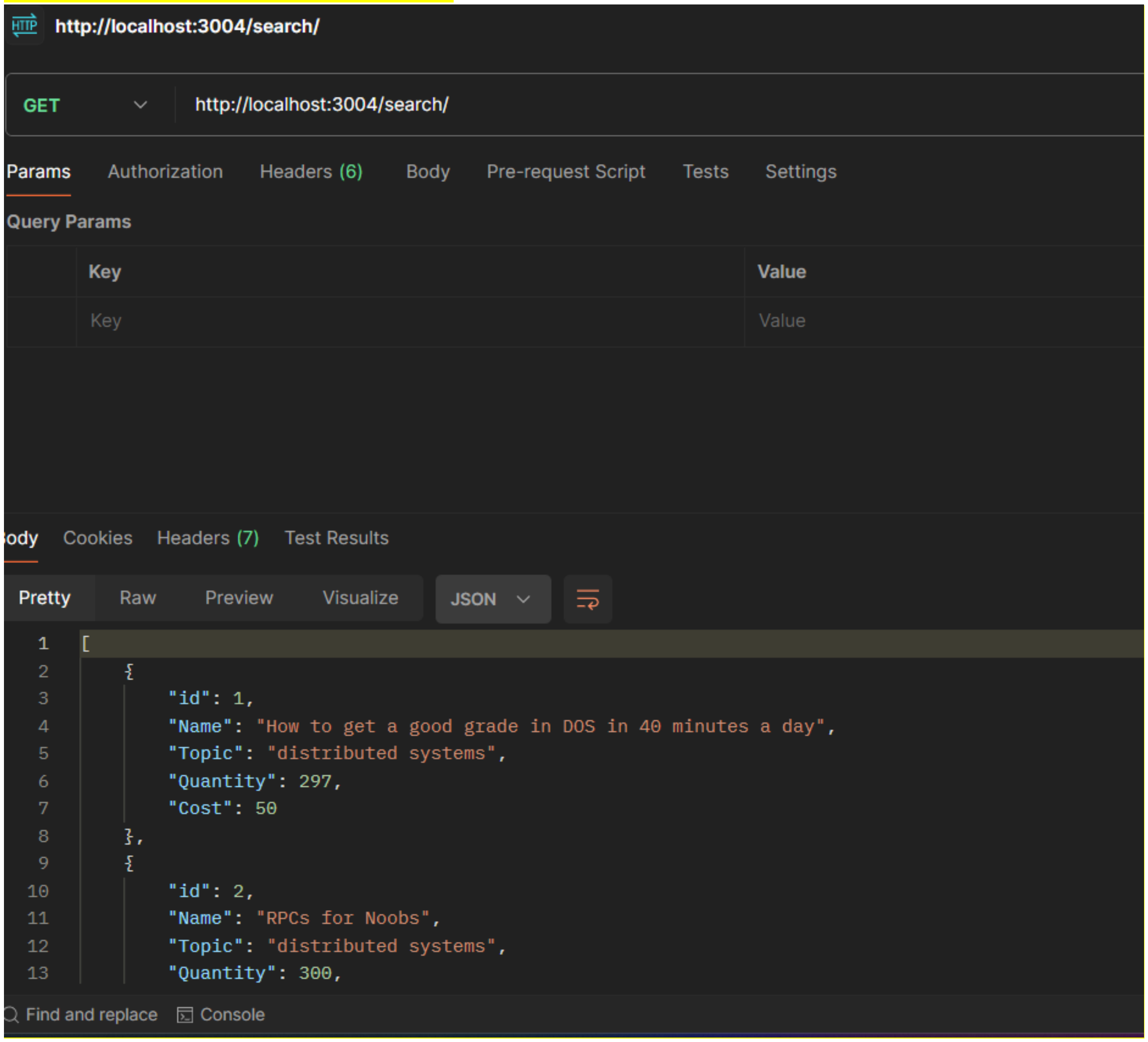
http://localhost:3004/info/1

GET http://localhost:3004/info/1

```
{
    "id": 1,
    "Name": "How to get a good grade in DOS in 40 minutes a day",
    "Topic": "distributed systems",
    "Quantity": 297,
    "Cost": 50
}
```

2-To search for all the books in the database, send

http://localhost:3004/search/

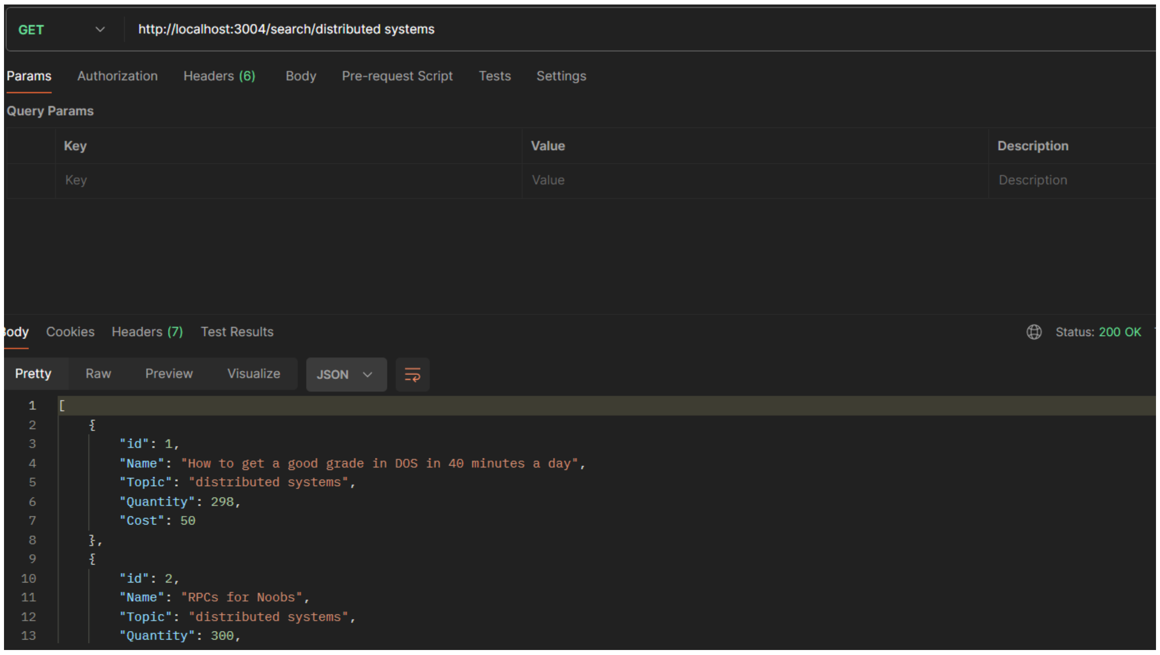


GET http://localhost:3004/search/

```
[
    {
        "id": 1,
        "Name": "How to get a good grade in DOS in 40 minutes a day",
        "Topic": "distributed systems",
        "Quantity": 297,
        "Cost": 50
    },
    {
        "id": 2,
        "Name": "RPCs for Noobs",
        "Topic": "distributed systems",
        "Quantity": 300,
```

3-To search for a book by it’s Topic in the database, send

http://localhost:3004/search/{TopicName}



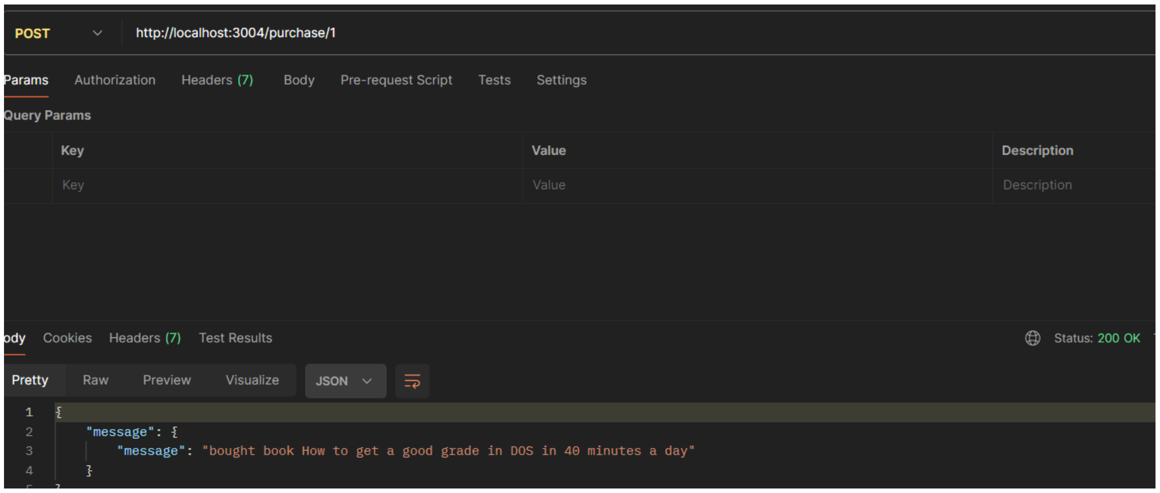
```
GET http://localhost:3004/search/distributed systems
```

Status: 200 OK

```
[
    {
        "id": 1,
        "Name": "How to get a good grade in DOS in 40 minutes a day",
        "Topic": "distributed systems",
        "Quantity": 298,
        "Cost": 50
    },
    {
        "id": 2,
        "Name": "RPCs for Noobs",
        "Topic": "distributed systems",
        "Quantity": 300,
```

4-To Buy some book’s in the database, do this command with Post request:-

http://localhost:3004/purchase/1



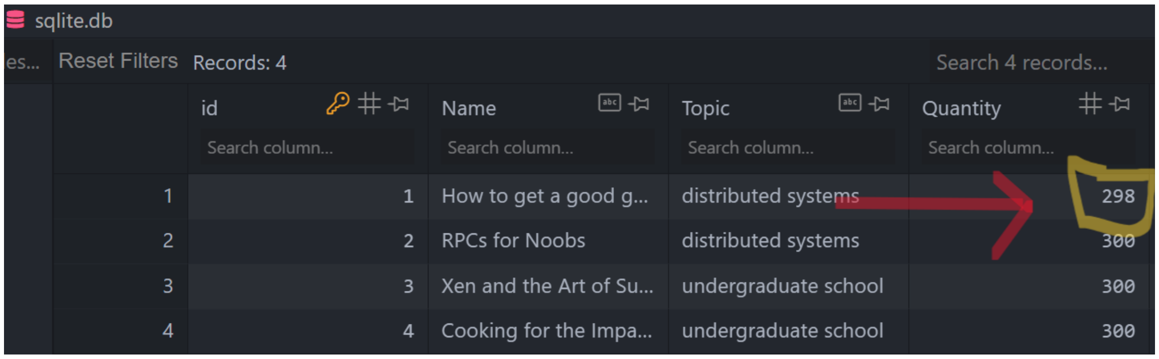
```
POST http://localhost:3004/purchase/1
```

Status: 200 OK

```
{
    "message": {
        "message": "bought book How to get a good grade in DOS in 40 minutes a day"
    }
```

For the purchase order this is the result and the difference in the database before and after this request:-

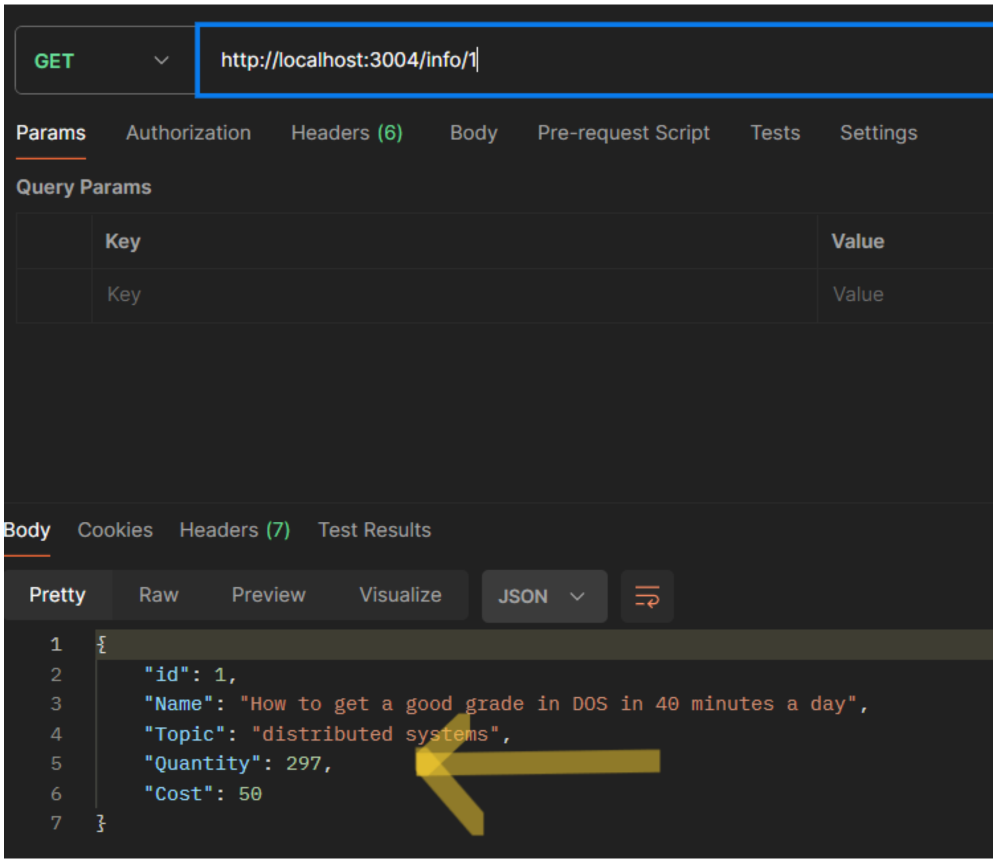
Before:-


sqlite.db

Reset Filters Records: 4

	id	Name	Topic	Quantity
1	1	How to get a good g...	distributed systems	298
2	2	RPCs for Noobs	distributed systems	300
3	3	Xen and the Art of Su...	undergraduate school	300
4	4	Cooking for the Impa...	undergraduate school	300

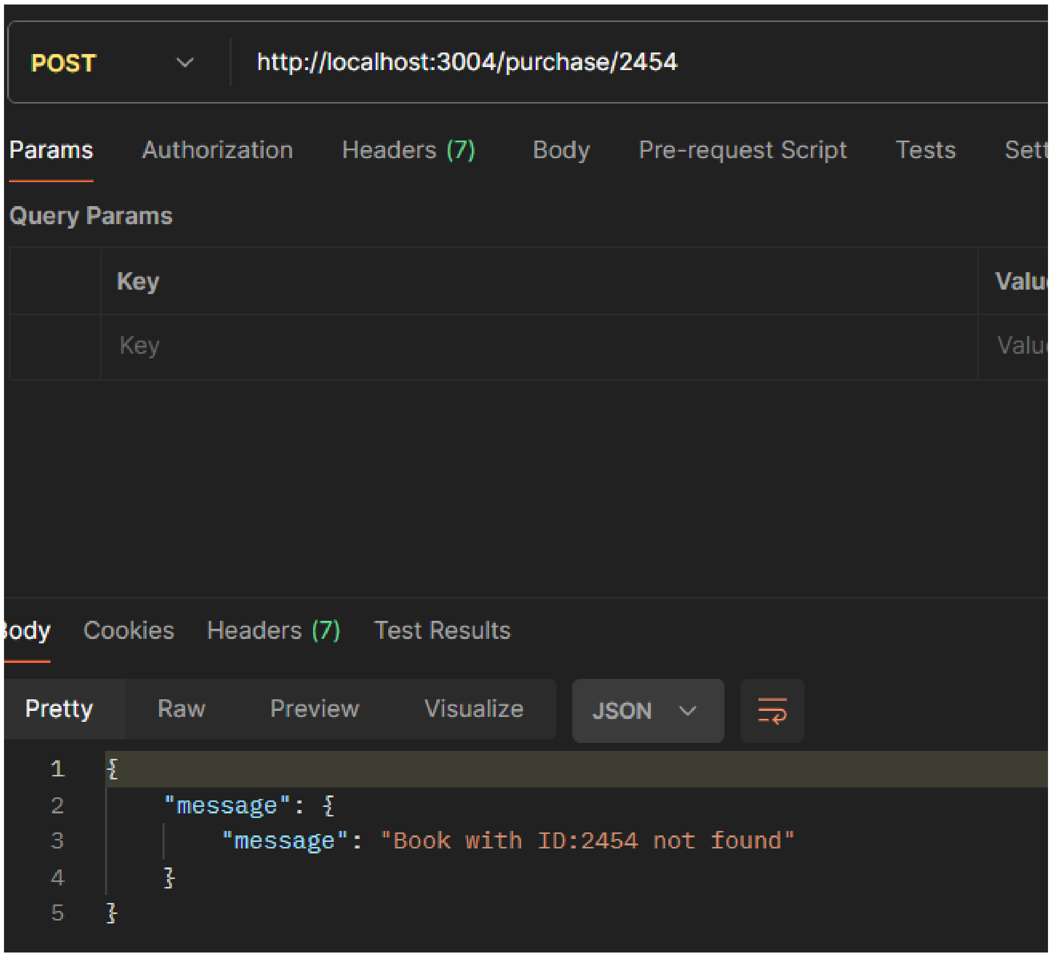
After:-


GET http://localhost:3004/info/1

```
{
    "id": 1,
    "Name": "How to get a good grade in DOS in 40 minutes a day",
    "Topic": "distributed systems",
    "Quantity": 297,
    "Cost": 50
}
```

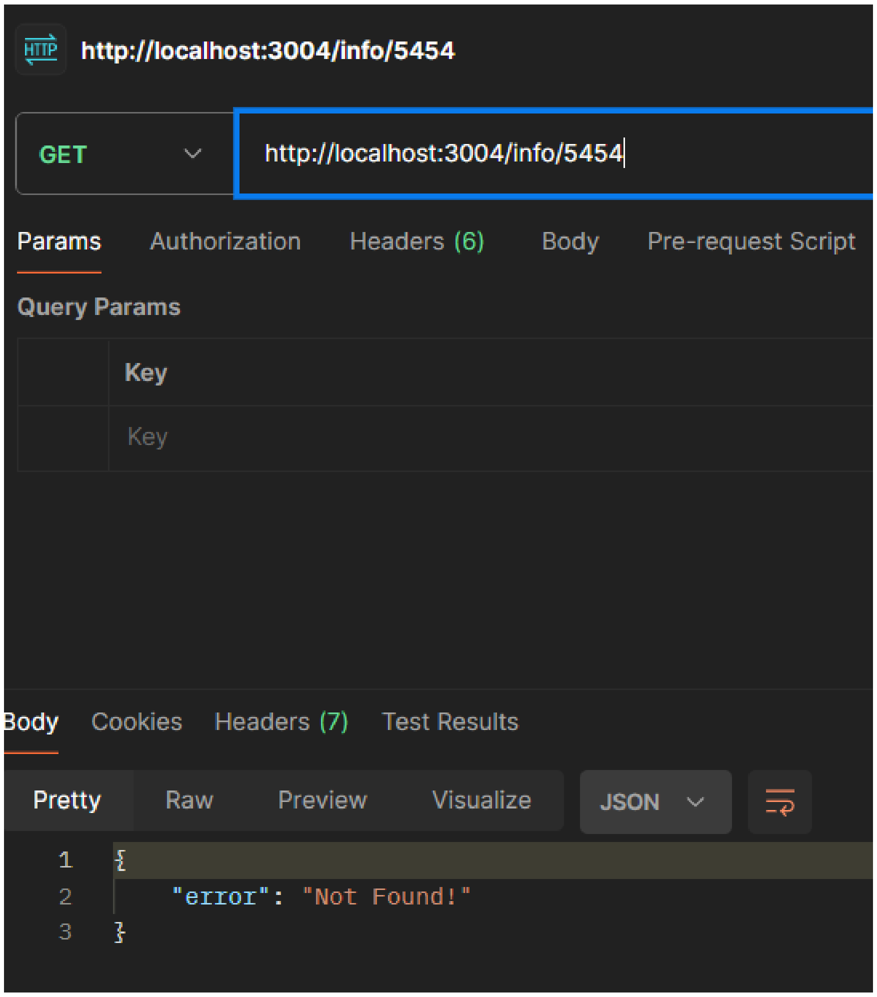
For each request we solve all the cases like: some book's not found, or there book with this id, also for purchases we did some check on the count on the database, so if there is no more it will show a message for that case.

These are some output's for these cases:-



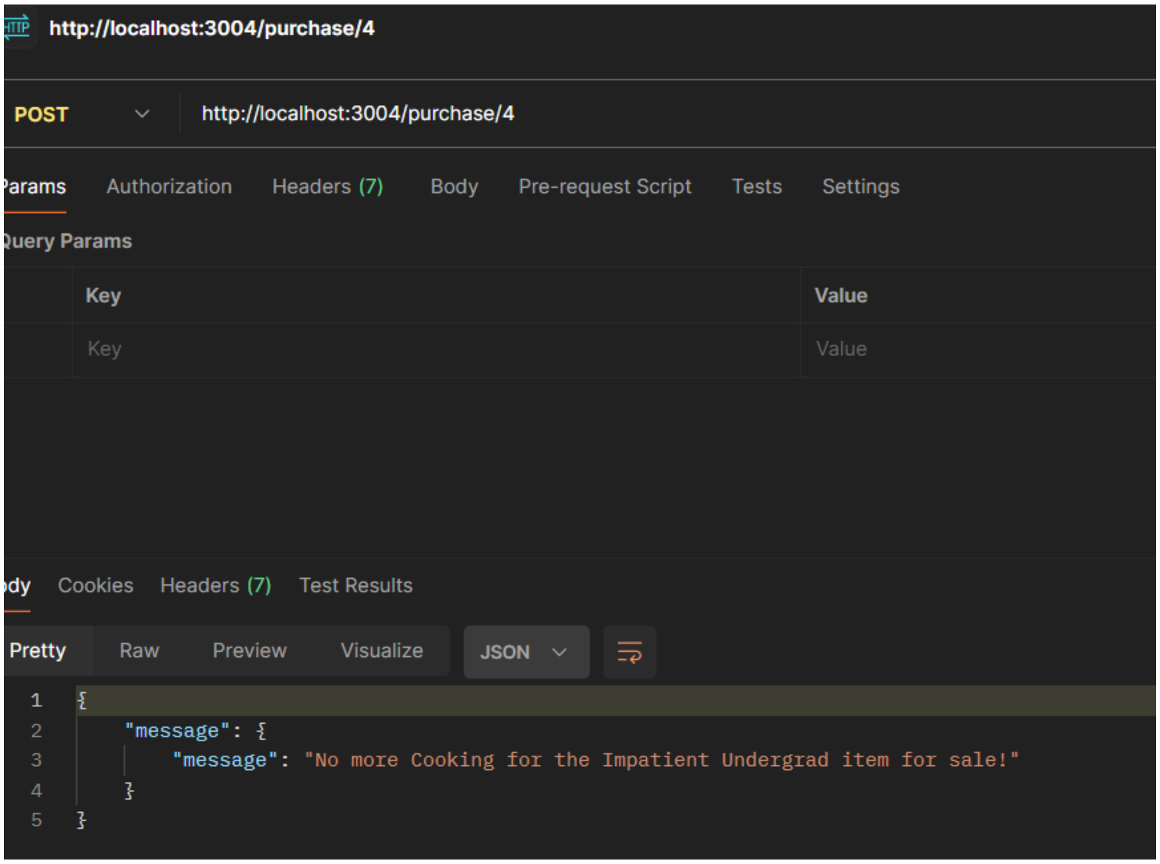
http://localhost:3004/purchase/2454

```
{
    "message": {
        "message": "Book with ID:2454 not found"
    }
}
```



```
http://localhost:3004/info/5454
```

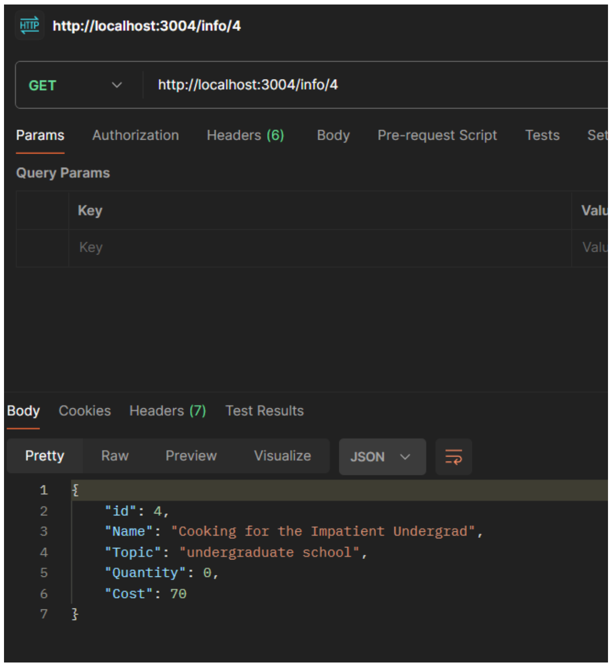
```
{
    "error": "Not Found!"
}
```



POST http://localhost:3004/purchase/4

```
{
    "message": {
        "message": "No more Cooking for the Impatient Undergrad item for sale!"
    }
}
```

This is the case such as there is no more books in the database!



GET http://localhost:3004/info/4

```
{
    "id": 4,
    "Name": "Cooking for the Impatient Undergrad",
    "Topic": "undergraduate school",
    "Quantity": 0,
    "Cost": 70
}
```

Problems

In some cases sometimes the Docker desktop get crashed, may be because of the pressure on the servers, this is the only issue we faced when we sometimes run the Docker containers.